AD-E501733

UNCLASSIFIED

Copy 16 of 50 copies

2

IDA PAPER P-2636

SOFTWARE RISK ASSESSMENT FOR DoD ACQUISITION PROGRAMS

Dennis W. Fife, *Task Leader*

Bill Brykczynski
Beth E. Springsteen

September 1993

Approved for public ralease, unlimited distribution: November 1, 1993.

IDA

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

UNCLASSIFIED

IDA Log No. HQ 93-044532

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this document was conducted under IDA's Central Research Program. Its publication does not imply endorsement by the Department of Defense or any other Government Agency, nor should the contents be construed as reflecting the official position of any Government Agency.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1993	3. REPORT TYPE AND DATES COVERED Final

4. TITLE AND SUBTITLE
Software Risk Assessment for DoD Acquisition Programs

5. FUNDING NUMBERS
IDA Central Research Program (CRP) 9000-530

6. AUTHOR(S)
Dennis W. Fife, Bill Brykczynski, Beth E. Springsteen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Institute for Defense Analyses (IDA)
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-2636

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311-1772

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, unlimited distribution: November 1, 1993.

12b. DISTRIBUTION CODE
2A

13. ABSTRACT (Maximum 200 words)

This is a guide for assessing computer software aspects of major defense system acquisition programs. IDA's methodology for software assessments is based on experience, and has been documented and refined through internally funded research. An assessment project reviews processes and resulting software products of a DoD acquisition program office and its contractors. Assessment is done by a technically well-qualified team that is independent of the program office. Results inform the program manager of problems and risks that threaten successful software delivery and also assist risk reduction decisions. This guide is provided to assist any DoD sponsored software assessment team. The Overview chapter summarizes IDA's methodology for program managers and acquisition executives who may need assessment assistance. Individual chapters define assessment phases and recommendations for conducting them. A checklist is given for identifying existing software problems and risks within an acquisition.

14. SUBJECT TERMS
Software Risk, DoD, Software Acquisition, Acquisition Risk, Program Review, Program Assessment

15. NUMBER OF PAGES
90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR

IDA PAPER P-2636

SOFTWARE RISK ASSESSMENT FOR DoD ACQUISITION PROGRAMS

Dennis W. Fife, *Task Leader*

Bill Brykczynski
Beth E. Springsteen

September 1993

Approved for public release, unlimited distribution: November 1, 1993.

INSTITUTE FOR DEFENSE ANALYSES
IDA Central Research Program
9000-530

PREFACE

This document fulfills the objective of the Institute for Defense Analyses (IDA) Central Research Project 9000-530, to provide a guide for expert teams to assess the software aspects of Department of Defense acquisition programs.

Dr. Harlow Freitag and Mr. Robert J. Knapper participated in the project at an early stage, and Mr. David A. Wheeler made significant suggestions which are embodied in this report.

The following IDA research staff members reviewed this document: Mr. John N. Donis, Dr. Cy D. Ardoin, Mr. David A. Wheeler, Dr. Judy Popelas, and Mr. Terry Mayfield.

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

This is a guide for assessing computer software aspects of major defense system acquisition programs. The Institute for Defense Analyses (IDA) methodology for software assessments is based on experience, and has been documented and refined through internally funded research. An assessment project reviews processes and resulting software products of a Department of Defense (DoD) acquisition program office and its contractors. Assessment is done by a technically well-qualified team that is independent of the program office. Results inform the program manager of problems and risks that threaten successful software delivery and also assist risk reduction decisions.

This guide is provided to assist any DoD-sponsored software assessment team. The Overview summarizes IDA's methodology for program managers and acquisition executives who may need assessment assistance. Individual chapters define assessment phases and recommendations for conducting them. A checklist is given for identifying existing software problems and risks within an acquisition.

DoD acquisition executives and program managers need to assess software status because computer software is a major factor in acquisition programs. The DoD spends tens of billions of dollars per year on software development and support, although sources may differ on a specific figure [EIA 1989, Kimmel 1993]. Technical factors, complexity, and scale of software developments often are unfamiliar to DoD program managers and their staff. Thus software risks to program cost, schedule, system performance, and supportability often go unassessed and unmanaged. Effective assessment depends upon appreciable software experience for which military acquisition standards and guidebooks are no substitute.

IDA's approach evolved primarily from an assessment project done on the AN/BSY-2 Submarine Combat System, originally done at the request of the Director, Naval Warfare and Mobility, in the Office of the Deputy Director of Defense for Acquisition, Tactical Warfare Programs. Another influence was IDA experience in assessing contracted

software development using Software Engineering Institute (SEI) process maturity methodology. This guide includes further information about these experiences.

OVERVIEW OF SOFTWARE ASSESSMENT FOR CUSTOMERS AND NEW TEAMS

This overview summarizes the Institute for Defense Analyses (IDA) software assessment methodology for Department of Defense (DoD) acquisition executives or program managers (PMs) who may sponsor assessments and for analysts and engineers who are newly assigned to perform assessments.

The main chapters of this guide address goals and techniques for conducting each phase of an assessment project. The purpose of the guide is to help software assessment teams proceed rapidly with a proven approach. The guide includes key technical references and experience reports from prior assessment projects.

PURPOSE OF SOFTWARE ASSESSMENT

Software assessment provides *DoD acquisition executives or PMs with an independent* and objective view of ongoing software development in an acquisition program. The approach identifies software risks, assesses their potential impact on the acquisition, and provides risk mitigation recommendations for the PM. The results of an assessment are primarily used to assist decisions to undertake risk reduction action.

CONTEXT OF AN ASSESSMENT PROJECT

This guide may serve various types of assessment projects, but for convenience, the primary interested party and funding source will be taken to be a high-level Service acquisition executive or an acquisition PM. An assessment project reviews the software aspects of a defense system acquisition, hereafter called the system or acquisition under review. The system acquisition is being conducted by a program office (PO) within an acquisition agency of DoD or the military services. A government PM heads the PO and has overall responsibility for the program, including risk management. The system's software is being developed (or integrated and tested, if off the shelf) and delivered by a development contractor. The software development contractor may be the prime contractor for the system in which the software will be embedded, a key subcontractor of the prime contractor, or even a government development agency.

Software assessment as advocated here is performed by an independent team that provides exceptional technical knowledge and broad experience in software development, and a measure of objectivity. Software assessment includes review of the PO activities and results, the development contractor effort, and software products delivered to the PO. The acquisition under review already may have an independent validation and verification (IV&V) contractor or other support providing the PM with technical reviews and risk assessments. Even so, the recommended external team review will be valuable for its objective comparison with the PM's risk assessments, stronger software knowledge and analysis, or insight into improving the PO's effort.

"Task leader" refers to the individual responsible for technical direction of the assessment project team. The task leader negotiates necessary agreements, coordinates with the PM and sponsor (if other than PM), development contractors, and interested OSD parties if appropriate, and acts as the team's primary interface with the PO and development contractor.

PRINCIPAL ASPECTS OF IDA'S APPROACH

This section summarizes important attributes of IDA's approach, defines terms, and gives rationale for the methodology.

Risk Perspective

Acquisition assessments could be undertaken with various points of view or goals in mind. For example, the DoD often is interested in how well a program is advancing a technology that could serve other future defense needs. Another frequent interest is documenting lessons learned about specific acquisition or development issues. IDA's approach centers on a comprehensive assessment of software development risks within a given acquisition.

The motivation is to provide acquisition managers with maximum information for achieving program success and avoiding runaway projects, costly rework, or improper focus on low-leverage activities.

The term "risk" warrants a brief explanation. A risk is a subject, topic, or aspect of software development that could, if certain events or conditions occur, produce negative consequences such as cost overrun, schedule slippage, or reduced product or system performance. A problem, on the other hand, is a risk that has materialized, i.e., the negative events or conditions are observable facts and the impacts are in progress. Glibly put, *a problem is a risk whose time has come.*

Risk thus connotes a degree of uncertainty about whether an issue actually will emerge as a problem. It is entirely possible for an acquisition to progress with many persistent high risk issues and to finish successfully with no serious problems ever coming to pass. Identifying risk issues that do not in fact emerge as problems, or that cannot be eliminated for practical reasons, is not bad risk assessment. Bad risk assessment is ignoring or giving low priority to an issue that soon erupts as a major problem or barrier for acquisition success.

Adaptable Scope and Duration

IDA recommends a comprehensive assessment that addresses any software-related issue that could significantly impact the acquisition under review. Ongoing problems, if any, may be known already to a PO, and some of the risks may be recognized too. The assessment methodology provides an independent determination of the program situation, improves understanding of risk likelihood and impact, and recommends priorities and solutions for risk reduction. Risk identification and assessment must come first, but at the sponsor's option, an assessment project may extend into risk reduction planning.

An assessment examines a wide range of subjects and results that fall primarily into these four areas.

a. The development process and key practices of the software contractors.

b. The products being delivered by the software contractors.

c. The process of the responsible acquisition PO for directing and monitoring the overall program.

d. The key products of the PO.

Examples of contractor development practices include project planning, quality assurance, and risk management.

Examples of contractor products include the derived software requirements, software design, and source code. Most products will be documents conforming to the DoD software development standard, DoD-STD-2167A [DoD 1988], or its imminent successor [DoD 1992].

Examples of the PO process include acquisition planning and requirements management. Examples of PO products include requirements, risk management plan, and development status data.

This guide is intended for assessment that is focused almost entirely on software development. However, system engineering, overall program risk management, program quality assurance, and computer and communications hardware are closely related. For a given assessment, the negotiated scope may somewhat exceed the explicit bounds of this guide.

The allowable time duration of an assessment project primarily determines the depth to which risks and impacts can be evaluated. The methodology can be adapted to a very short project schedule, although a duration up to six months is preferable. The methodology is modular and flexible for meeting either schedule goals or sponsor-targeted priorities. An assessment could focus largely on product performance issues impacting mission effectiveness, on PO activities, on development process matters affecting software quality, or have a balanced perspective addressing all of these areas and others. In most cases, a comprehensive survey is preferable, so that no important risk is overlooked entirely.

Recurrent Assessments

Risks change substantially during the acquisition life cycle because of development progress, external events or conditions, and evolving requirements. All major acquisition programs need a continuing software risk management effort from the PO and contractors. Even so, the independent external review advocated in this guide is recommended for several time-points during a program's life. The independent team effort complements and helps calibrate program-internal risk assessment and management.

A first assessment close to Milestone 0, as Concept Definition results become available and Demonstration/Validation (DEM/VAL) plans are being formulated, concentrates on the software technology challenges of the program and their recognition in basic plans and requirements. Assessment stresses basic planning factors, such as technology insertion needs, realistic estimation of software development effort, and benefits from past experience. Assessment results help a PO formulate acquisition requirements, DEM/VAL software evaluations, and risk mitigation plans.

A second assessment near the conclusion of the DEM/VAL phase (before Milestone 1) concentrates on the technology results and plans evolving from DEM/VAL, including software standards, prototyping conclusions, reuse opportunities, and development methods, technology, and tools. Significant software devtlopment, especially of tools and prototypes, may be performed by contractors during DEM/VAL. Available contractor experience and results, e.g., software reuse planning, will point out further action needed to mitigate risk. This assessment is especially important if the acquisition plan includes substantial reuse of DEM/VAL software in later program phases.

One or more assessments during Engineering and Manufacturing Development (EMD) concentrate on software development, integration, and testing progress, and evolution of risk mitigation plans. Software development may be largely completed during EMD, but heavy effort in system testing near the end of EMD may expose increased risk or serious problems in unsatisfied requirements.

In regard to software, the Production phase of acquisition is likely to focus primarily on software product improvements and post-deployment software support (PDSS) needs. Additional risks are typically raised in the transition between original developers and any separate support organizations.

Cross-disciplinary Approach

IDA's approach calls for an assessment team to include both software and application experts. This recognizes that a robust assessment depends upon experience with the mission requirements of the system under review as well as a broad range of software development knowledge. For example, in IDA's BSY-2 assessment the software experts received input from submariners and sonar experts who knew the operational and performance risks inherent in the system. These "domain experts" contributed in many ways, assisting in data gathering, impact analysis, and communication of the findings to PO staff. Further, the software cadre included experts in Ada development, data communications software, software testing, computer architecture and operating systems, and other specialty areas within software engineering.

ASSESSMENT ACTIVITIES

A risk assessment project for any of the recommended points during acquisition involves the consecutive phases depicted in Table 1. These phases and their activities are summarized next, then detailed in subsequent chapters.

Preparing for Assessment

The first phase of an assessment establishes the assessment plan and team membership. It begins with agreement between the task leader and task sponsor on the scope of the assessment. The task leader provides the sponsor and PM with an overview of the assessment approach, including the methods, support, and resources needed. The task sponsor and PM, by identifying their objectives and schedule constraints, help to focus the project and set its priorities.

The task leader selects assessment team members and their assignments, and lays down additional detail for the project schedule. The team obtains necessary project orientation, including status briefings on the acquisition under review. The collection and review of development documents begins.

Table 1. Phases and Activities of an Assessment Project

Principal and Subordinate Activities
Preparing for assessment • Scoping the project • Scheduling the project • Staffing the team • Reviewing the program • Requesting documents
Collecting risk data • Reviewing documents • Finding candidate risks • Conducting site visits • Evaluating contractor process maturity
Assessing risk • Ranking a final risk list • Refining risk impact and indicators
Reporting findings • Briefing findings • Documenting findings
Mitigating risk

Collecting Risk Data

The data collection phase centers on gathering facts and evidence for identifying and assessing risks for the acquisition under review. Relevant contractor and PO documents are acquired and reviewed. Results from prior assessments, if any, are studied. Relevant experience from other development programs and commercial practice is identified. Close communication is established with the program's contractors, and site visits are planned and conducted to interview their software development personnel. A preliminary list of candidate risks is assembled using this guide's risk checklist found in Appendix A.

Assessing Risk

In this phase, the assessment team jointly reviews preliminary findings and integrates information bearing on the likelihood and potential impacts of the candidate risk issues. Risk assessment involves evaluating three factors: (1) the likelihood of a risk becoming a problem, (2) the impact in cost, schedule, and performance if it does become a problem, and (3) the necessary investment cost and lead time to minimize the impact should a problem emerge. Suggestions are developed on preparatory actions that would reduce or mitigate each risk. This analysis produces the principal findings on significant problems and risks, and the supporting information on impacts and risk reduction.

Reporting Findings

Findings from an assessment are expected to be delivered through a briefing given to both the task sponsor and PO personnel. Findings include positive program achievements relative to software risk, analysis of significant risks as to their potential impact, and recommendations for risk reduction. If an archival report is needed, the briefing charts can be augmented with explanatory text and documented background material.

Mitigating Risk

During the risk identification and analysis phase of an assessment, ideas and suggestions will emerge for readily reducing or eliminating some of the risks. Other identified risks will require in-depth analysis in order to develop risk mitigation plans. The last, optional phase of software assessment involves the assessment team working with the PO to develop in-depth mitigation plans for high priority risks.

USE OF THIS GUIDE

The rest of this guide is primarily for the task leader and members of an assessment team. It outlines an assessment project, recommends a number of principles and practices, and serves as basic orientation to a recommended risk assessment approach. Appendices B and C recount lessons from past IDA projects.

This guide is not conceived as an exhaustive and rigid prescription for every project. It is a starting point to help quickly formulate the approach for a given project. In particular, specific assessment activities and key issues for a given project will depend upon the acquisition status and evolution of the program under review. This guide is written mostly for assessments done near the end of DEM/VAL or early in EMD. For other points in the acquisition life cycle, an assessment team may have to shift the emphasis of the checklist and review guidance. For example, an assessment during Concept Definition would emphasize technology issues more than contractor development processes. Thus the product aspects of the checklist might need expansion, rather than the process criteria. Program results would likely be seen in general technical and engineering reports rather than formal deliverables made in compliance with the DoD-2167A software development standard.

As a team member gains experience by performing assessment projects, this basic guidance will become familiar and not needed for daily reference. However, during an ongoing assessment, team members may find the risk checklist, Appendix A, to be a helpful reminder of issues that need consideration.

1. PREPARING FOR ASSESSMENT

This chapter provides specific guidance on activities and arrangements for beginning an assessment project. For generality, contracting and funding for the project are not covered. The assumed starting point is after the requisite commitment is made and substantive technical interactions can begin with program office and contractor staff.

1.1 SCOPING THE PROJECT

The task leader meets informally with the program manager and the sponsor (if not the PM) to resolve the assessment project's scope and to exchange basic information on assessment methodology and the system under review. The task leader briefs the assessment approach to the sponsor and program manager, covering its potential benefits to each. Establishing cooperation, communication, and mutual understanding of the assessment's scope and potential results are very important. Specific examples drawn from past project experiences will help reach concrete understanding of the team's approach.

The scope and depth for an assessment may be driven by several factors including schedule or funding goals. Factors that pose difficulty or need special attention must be identified, such as program office cooperation, mission and application complexity, program status, or security classification and access issues. For completeness and technical precision, a comprehensive review is recommended, as was done for BSY-2, see Appendix B. This may take four months or longer, but provides the most certain data on a program's status and its current or impending problems. Sponsor and program manager may have risk reduction goals that require further analysis beyond the basic level needed to identify critical risks and existing problems.

If only a minimal project is acceptable to the sponsor, then establishing a well-delineated objective is crucial. The standardized review known as Software Capability Evaluation (SCE)[1] provides one example. It gives a tightly focused approach that addresses only the contractor's development process. Other factors that may serve to delineate scope

[1] Appendix A includes the issues examined as potential risks in the SCE methodology, and Appendix C briefly describes Software Capability Evaluation practice per IDA experience.

include whether or not program office activities and products are addressed, whether or not contractor products are assessed, or whether or not certain risk areas are reviewed.

1.2 SCHEDULING THE PROJECT

A candidate project schedule can be drafted by assigning equal work intervals or time units to the phases in Table 1. Then adapt and refine this based on the needed duration for each assessment phase and specific dates for key events in the project. Although tightly scoped projects can be performed on a schedule as short as a few weeks, a task leader needs to be conservative about delays to schedule appointments, acquire documents, and accomplish other externally-controlled activities. Also, many activities necessarily involve most or all of the team, so be conservative about planning concurrent efforts by many team members.

1.3 STAFFING THE TEAM

The assessment team will include members with significant background in the type of system under review. These domain experts ideally have direct experience with the mission activity involved and personal knowledge of the acquisition agency responsible for the program. Domain experts participate on an equal footing with software experts. They have in-depth knowledge of the role of computer resources in providing system capabilities, and of typical functional allocations to computer hardware and software. Their knowledge encompasses typical system algorithms and operating concepts, and the relationship of allocated computer functions and performance to overall mission capability. They assist in identifying and examining particularly difficult portions of the system being developed. Domain experts help assess product quality, relate system requirements to software functionality, identify system engineering and program planning weaknesses that may affect software, and help present risk assessment findings to program office or other acquisition staff. They also help the team tap into other DoD information sources in order to investigate precedents for the system under review.

Software experts on the team will cover important specialties that are involved in the system under review, such as computer networking and data communications, Ada software development, real-time system design, human-system interfaces, etc.

Team members will have sufficient education and experience that no formal training is necessary for beginning an assessment project. Basic references such as [Boehm 1989], [AFSC 1988], and [DoD 1988] or [DoD 1992] should be familiar to all team members. If the team members are not well acquainted with one another from common past projects, the team should meet to review the methodology and share prior experience and lessons learned.

It may be effective to assign assessment team members according to the major functional areas found in the risk checklist, Appendix A, or by groupings of the areas. For example, one team member might be responsible for examining all potential risks associated with contractor quality assurance. That would encompass reviewing all documents that deal with quality assurance and developing and asking questions regarding quality assurance during site visits, as well as performing the risk analysis for this area.

An important staffing consideration is forming a Software Capability Evaluation team either within the assessment team, or as adjunct to it. This guide advocates using an SCE as a baseline in any assessment, and it is helpful for consistent results to have an experienced IDA team perform the SCE of the contractor during the assessment.

One team member or a supporting administrative staff person should be designated to serve as team librarian. The librarian takes the lead in acquiring, tracking, cataloging, organizing, and disseminating system and program documents for benefit of the whole team.

1.4 REVIEWING THE PROGRAM

After the team is established, the task leader requests a briefing by the Program Office on the system under review. Briefings from supplementary sources also may prove helpful to fill out background on the system and its mission. The assessment project is underway once the team as a body has had the program overview and status briefing from program office and prime contractor staff. This briefing should include information such as system technical and operational factors, acquisition status, contractors and their responsibilities, and program office risk management results.

Determining the program's status in the acquisition life cycle is essential for the team to begin identifying the more likely risk issues for the contractor and the program office. For example, in the EMD phase, contractor quality assurance activities may have more risk potential than development tools and technology, since the latter should be well established and in daily use. Having said that, team members must be alert for any issue to actually be a major risk or problem in an unexpected or unaccepted way. Surprises more often are due to known events or conditions that are being ignored as unimportant than to completely unpredictable events occurring.

1.5 REQUESTING DOCUMENTS

As soon as feasible, the task leader requests the Program Office to provide documents on program activities and results to date. A suggested set of documents to request initially is listed below. Actually acquiring a sufficient, up-to-date set of documents may be a challenge for the team, and some delay would be typical.

For programs with software development well underway, much information will be in the form of required deliverable documents conforming to DoD-2167A or similar standards. The following identifies important contractor documents that should be acquired for review. Proper names and acronyms refer to documents defined in DoD-2167A, but equivalent or highly similar documents should be identifiable where other standards are being used.

a. Software Development Plan (SDP)

   A Software Development Plan describes a contractor's plan and methodology for software development.

b. Software Requirements Specification (SRS)

   A System/Segment Specification (SSS) provided by the government may contain many software requirements. But typically it is a contractor responsibility to develop complete and explicit software requirements derived from the SSS and stating technical considerations for meeting system requirements.

c. Software Design Documents (SDDs)

   Each Software Design Document (SDD) describes the complete design of a Computer Software Configuration Item (CSCI). A CSCI is the lowest level software component under configuration management.

d. Formal Review Materials

   Contractor presentations at reviews already held, e.g., Preliminary Design Review or Critical Design Review, provide effective summaries of approaches, problems, and issue resolutions in program management.

e. Software Test Plan (STP)

   A Software Test Plan (STP) describes the plans and software test environment required for formal qualification testing (FQT) of CSCIs.

f. Contractor's risk management plan

The contractor's risk management plan evaluates the known risks and describes how further risks will be identified, assessed, and mitigated.

The following information also should be requested initially to cover the Program Office effort.

a. Request For Proposals (RFP) (i.e., all RFP requirements and technical specifications such as the System/Subsystem Specification)

b. Awarded contractor's proposal, and the corresponding Statement of Work (SOW) and Contract Data Requirements List (CDRL)

c. PO risk management plan

d. Cost analysis requirements document

e. PO guidance to contractor from past reviews and delivered documents

f. Test and Evaluation Master Plan (TEMP)

2. COLLECTING RISK DATA

Effective assessment depends upon obtaining currently valid information about the program under review and using this information systematically to isolate potential risks from the milieu of ordinary technical and management activities. This chapter advises the assessment team on collecting information and using it to guide and focus assessment effort.

2.1 REVIEWING DOCUMENTS

Acquiring current and pertinent documents as quickly as possible is crucial to meeting any assessment project schedule. The task leader and team must be determined and persistent in order to successfully access the most relevant information, without becoming overloaded or diverted by marginally useful information. Helpful insight and cooperation from the program office and contractors take on immense value in this regard. Lack of cooperation may make a valid assessment unachievable. The task leader should monitor information gathering and dissemination by the team and guard against redundant information requests that tax program office and contractor support.

The point where an assessment occurs during the acquisition life cycle greatly affects the scope and content of available software information. The volume of material available in late DEM/VAL and EMD, including source code, likely forbids exhaustive examination. Initially the team should concentrate on the contractor and program office documents identified in the previous chapter, and look at high impact activities and products. The checklist, Appendix A, will help identify weaknesses and limitations of current program efforts and products. This initial document review should produce a list of prospective risks, notes about the pertinent facts or evidence, and questions pointing to additional data needed to substantiate assessments. Later, see next chapter, more document review may be needed to organize complete data for ranking risks according to their impact.

2.2 FINDING CANDIDATE RISKS

IDA experience indicates that an assessment team can find ongoing problems and potential risks through four types of analysis acquisition program and its software development. These are assumptions, exceptions, products, and processes for both the program office and the contractor team.

2.2.1 Assumptions Analysis

An assumption is any expectation or understanding for which there is no firm justification in fact, experience, or analysis. Assumptions about the world external to a program and about events not under PM or contractor control are the primary concern. Some examples of dubious assumptions are

a. A national or international draft standard not yet approved is assumed to achieve commercial importance early within the program life.

b. Experimental demonstration of a new technology is assumed to catalyze sweeping change of operational users' practices.

c. A community not involved in an acquisition is assumed to be an eager market for the system, although they have no compelling incentives to want it.

Various program participants might reveal important assumptions, which often are unstated or even unrecognized. The assessment team may need to ask many "how do you know?" and "what if it doesn't happen?" questions to draw out the dependencies and consequences.

2.2.2 Exceptions Analysis

Exceptions are highly unusual or unprecedented development situations, activities, goals, or reo irements. Remarkable departures from experience and good practice often are sources of program vulnerability. For example, is a contractor involved in modifying or enhancing commercial off-the-shelf (COTS) software from another vendor? Or, was the computer hardware selected before any significant analysis had been done of the software requirements? Or, does software development depend upon computer hardware that is still under development, not off-the-shelf? Also look at the contractor's experience in implementing solutions to requirements that are similar to the acquisition under review. Is this program unprecedented in either the contractor's or industry experience?

2.2.3 Processes Analysis

Modern quality management or statistical quality control has the view that a product's quality is highly dependent on the quality of development, manufacturing, and other processes that deliver it [Humphrey 1990]. The acquisition program office and the software development contractors both contribute to the overall process that delivers software. Thus the checklist in Appendix A covers both contractor and program office processes. This checklist can be an effective tool to identify potential risk issues. Figure 1 shows how it is organized and the individual topics within each category.

As illustrated, development process and software product topics form two categories, divided further into categories for contractors and program office. These in turn are decomposed into individual topics or subjects for investigation. A concise set of basic criteria for each subject is given to identify a low risk program. The assessment team must look for significant departures from these criteria in the program under review.

Other published checklists are available that can supplement Appendix A. [Donis 1993] is recommended for its complete life cycle analysis of DoD acquisition documents as an approach to finding risks. (It is intended for direct use by DoD acquisition managers, rather than a software-expert team, as is this guide.) [Carr 1993] and [JPL 1987] also have software-focused checklists. For an overview of process and product risk considerations emphasizing hardware and the late stages of the acquisition life cycle, consult [DoD 1985].

2.2.4 Products Analysis

Another means of finding risks is to compare a program's status and results with typical expectations for any program at the same state of evolution. For example, a team might use DoD standards and experience to develop specific criteria for what a program should have accomplished at its claimed life cycle stage. For a program near the end of the DEM/VAL phase, questions such as the following could be asked.

a. Has a software architecture been defined, with buildable CSCIs associated with each platform or subsystem (e.g., operations center, weapon, sensor)?

b. Have the critical algorithms for the mission been thoroughly defined and assessed, to understand the functional and performance attributes necessary to meet system effectiveness goals?

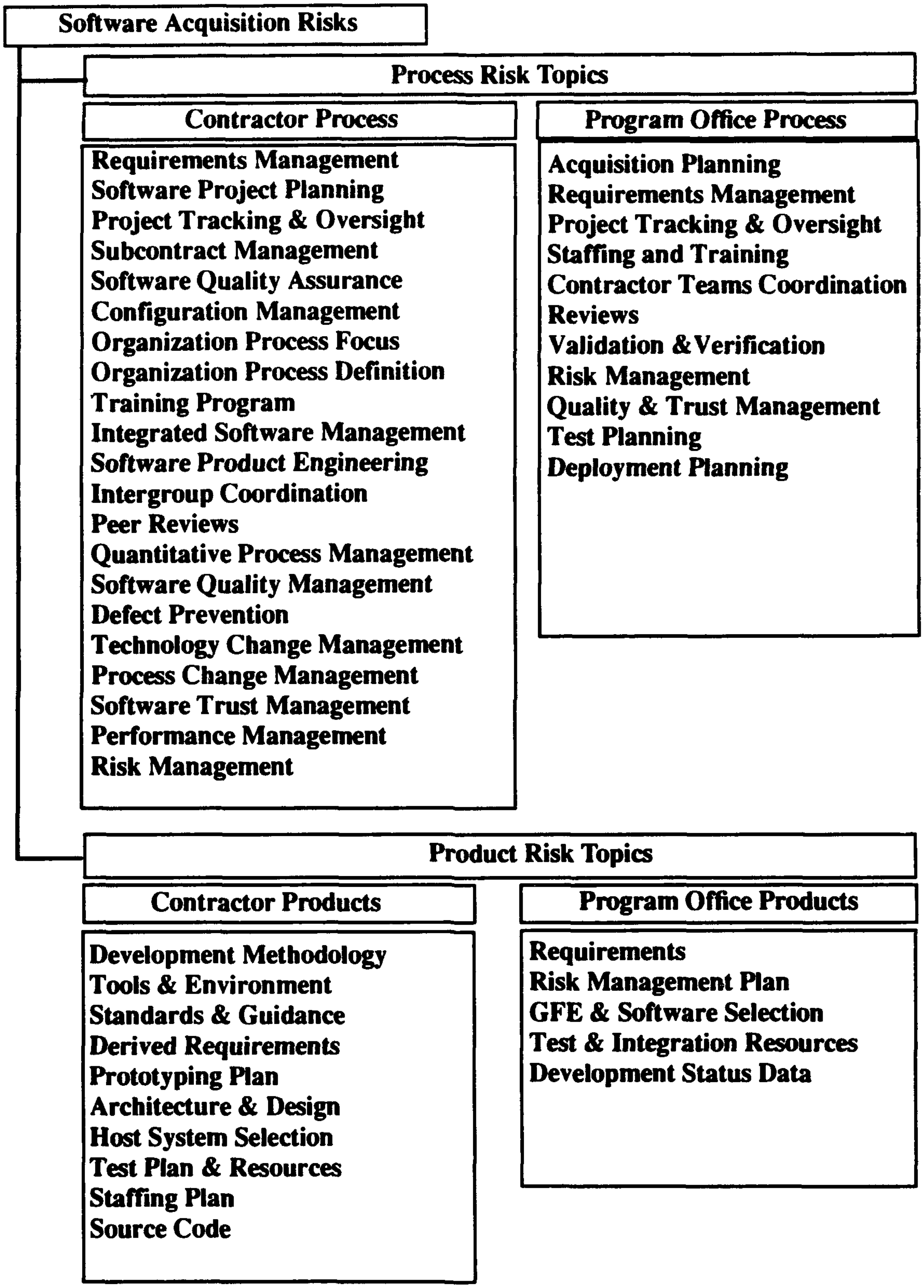


Figure 1. Risk Checklist Structure

c. Is the planned level of software reuse justified by an analysis of algorithms, operational functions, and performance for the intended reusable code in relation to the system's software requirements, architecture, and operations?

Appendix A product topics will help such analysis. Earlier plans and commitments in the program also should be revisited as of the assessment time. Have these expectations been fully realized? If not, a continuing and growing risk may be present.

2.3 EVALUATING CONTRACTOR PROCESS MATURITY

Weaknesses of contractors' software development processes, inexperience in performing the defined process, and inconsistent or haphazard conformance are all potential sources of risk to program performance and product quality. The recommended baseline method for identifying development process risks is the SEI process maturity model and the associated Software Capability Evaluation (SCE). The SEI model considers prior experience as a predictor of future performance and focuses on an organization's institutionalized or standard process for software development. The Appendix A checklist incorporates the current SEI model [Paulk 1993] and supplements it with other topics of concern to DoD acquisition programs. For example, trusted software development techniques are included in the risk checklist.

A Software Capability Evaluation is an important part of an assessment. For consistency of assessment findings, and to ensure that the program under review has been covered, it is preferable for an assessment team to perform a new SCE before its primary site visit to the contractor. However, using a prior SCE report may be acceptable if the assessment schedule or resources make a new SCE very difficult. It is essential however that the contractor organization was evaluated recently and by an independent government team, with those findings made available to the assessment team.

2.4 CONDUCTING SITE VISITS

Site visits should be made to the software development contractors in order to interview lead personnel and obtain a wide range of undocumented information needed for assessment. The prime system contractor and the major software development subcontractor (if different from the prime) should be visited at least. Site visits to other contractors are in order as well if their software components seem to pose risks.

All assessment team members should prepare for and participate in the visits, and a team coordination meeting before the visit is recommended. Team members should prepare by listing the major questions and evidence they need, and what contractor personnel (perhaps only by role, rather than name) they envision can provide answers. Coordination should eliminate duplication and may expose uncovered areas or issues.

In scheduling each visit, the task leader should provide the PO and contractor points of contact with a list of representative areas and questions that the assessment team wants to discuss. These should be based on the preliminary risks and data needs identified from the document review and from other input, e.g., the SCE report.

A full day should be allocated for each site visit. Additional time may be necessary for detailed analysis in specific areas, done by one on one discussion between a team member and contractor staff. Follow-up visits to the prime contractor or major software developer may be found desirable later in the assessment project.

Other site visits, e.g., to any IV&V organization or to producers/vendors of government furnished equipment or COTS software components also may be warranted by early risk indications.

3. ASSESSING RISK

This chapter addresses techniques for producing a final, ranked list of risks and the supporting evidence and findings which comprise the project's final briefing and report.

As stated earlier, risk identification and analysis begins as soon as program and system information becomes available. During document review, site visits, and other early activities, risk identification is tentative. The assessment effort concentrates on assimilating program status information and making informal judgements of the relative significance of many potential risks. This chapter concerns later risk analysis effort occurring after an assessment team has gathered the body of facts about the acquisition under review and now must draw well-supported conclusions about the most important risks.

3.1 RANKING A FINAL RISK LIST

The assessment team's broad exposure to a program's activities and products will enable it to produce perhaps a very long list of candidate risks. Those that looked significant at the outset may persist after site visits, but with different perceptions of their importance among team members. The team begins the later part of its effort with the need to coordinate their data and judgements, and to reduce potential risk issues to a manageable number, surely less than fifty, that have greatest importance and practical risk mitigation methods available.

The most important risk issues are those with the highest expected impact on program success. Expected impact involves the probability or likelihood that the issue will emerge as a problem, and if it does, the extent of the impact on program success. The practicality and cost of risk mitigation also must be factored into the impact determination. For example, a high probability event becomes unimportant if its negative consequence is limited and quickly correctable by low-cost action at the time it occurs. On the other hand, a low probability event is very important if its negative consequence is cumulative, e.g., creates a schedule bottleneck, and the only apparent recovery requires a substantial preplanned investment, i.e., a heavy insurance cost.

An assessment team's concluding analysis should define by consensus,

a. each significant risk in specific terms related to system mission and the program's activities and events,

b. the scope and nature of the negative impact on the program, exclusive of any risk mitigation plan, if the risk in fact occurs as a problem,

c. the likelihood of the risk in fact occurring as a problem,

d. initial judgement of the feasible risk mitigation actions available.

With this data in hand, the team proceeds to rank the risks in importance, also by consensus or joint decision.

3.1.1 Qualitative Ranking

A team may find high risk issues for which no practical risk mitigation is evident. These should be separated into a special category. These are risks that a program manager must knowingly accept. For example, in the BSY-2 review [Donis 1990], the Enhanced Modular Signal Processor (EMSP) was such a high risk item. EMSP was a required component by law, and so Navy managers had no practical risk mitigation option available.

A qualitative method is expeditious for ranking the remaining risks and also sufficient for many assessments. A qualitative method basically is a systematic approach to comparing and sorting items on the risk list. This can be done by identifying categories and putting items into the categories without further ranking, by making pair-wise comparisons, or by selecting certain items as reference items and ranking all others in relation to them. IDA's BSY-2 assessment placed risks into two categories according to the team's consensus judgement of their importance. The most important, those having a major system effectiveness impact, were labeled as "critical risks." Others were simply tagged as "risks." For instance, the lack of performance benchmarks for Ada language features was named a critical risk, while weakness in manpower usage planning was listed simply as a risk.

3.1.2 Quantitative Risk Measurement

In theory, risk is quantifiable as the probability of a problem occurring multiplied by the cost impact of the problem on a program. In an ideal world, probability of problem occurrence and cost of impact would be determinable on appropriate scales, and a basic multiplication of the two would support straightforward comparison and ranking of all risk issues. DoD program managers often like to distinguish risk in terms of performance, cost, schedule, or support impact, see [AFSC 1988]. To assess and rank all risks in uniform terms would require a uniform cost measure applicable to these different kinds of impact, such as reduced product quality, delay in product delivery, or increase in development cost. Measuring likelihood or probability of occurrence in consistent terms also may be quite difficult across a diverse set of negative events.

Software risk specialists currently debate and offer alternative ways to establish rankings on quantitative grounds [SEI 1993]. Quantifying risk, and especially quantifying the costs for alternative actions that mitigate risk, may become necessary for risk mitigation planning. The next chapter will return to the subject from that viewpoint. The first approach that a team considers for ranking risks and expressing their significance should be a qualitative method, especially one that can serve as a precursor to a quantitative method.

A compromise approach that is qualitative yet offers quantitative interpretation is suggested in [AFSC 1988]. It proposes using identified risk drivers that state alternative conditions or situations relative to performance, support, cost, and schedule risk categories. A table for each risk category indicates what probability level should be perceived from the identified risk driver conditions. For example, this pamphlet identifies REQUIREMENTS DEFINITION and REQUIREMENTS STABILITY as among 14 schedule drivers (see the table, Figure 6-2 of the pamphlet). It states that a schedule problem frequently will occur (citing numeric probability values of 0.7 or greater) if the requirements definition is unknown, with no baseline, and the requirements are undergoing rapid or uncontrolled change.

An assessment team should consider adapting or adopting the AFSC approach in conjunction with a qualitative ranking method. Identified risk drivers should help establish consistency in team judgements of likelihood and impact used in producing risk rankings.

3.2 REFINING RISK IMPACT AND INDICATORS

In producing its final, ranked risk list, an assessment team may depend significantly on subjective judgements. This may be inescapable in regard to the likelihood of uncertain and human-influenced events. But a team must objectively define the negative program impact perceived for each risk as well as the conditions or events that foretell the impact as imminent, i.e., that a risk is turning into a problem. Impacts and problems would be stated specifically in terms of program plans, events, costs, system requirements or delivered performance, and other observables. This is essential for program management to understand the risk assessment findings in concrete technical and management terms.

IDA's BSY-2 findings covered this information through statements of the Potential Impact and Risk Indicator for each risk issue. A Risk Indicator was an observable event or data that PO staff could follow to detect increasing likelihood of a problem. Potential Impact was expressed as the most immediate technical and programmatic consequences, rather than attempting to envision their propagation to later program events or metrics. Although it was desirable also to state impact on mission effectiveness, deriving this impact was beyond the scope of the BSY-2 assessment.

4. REPORTING FINDINGS

The reporting phase of an assessment project is concerned with 1) developing and coordinating a briefing on the findings, and 2) producing the necessary archival documentation.

4.1 BRIEFING FINDINGS

Following is one suggested structure for the briefing.

a. Summary of purpose, scope, and motivation of the assessment project

b. Summary of findings, identifying existing problems and critical risks

c. Summary of the approach and key events in performing the review

d. Basic review of each problem and critical risk in turn:

   (1) Detailed statement of the risk or problem

   (2) Potential impact, pertinent facts, and root source of the risk

   (3) Risk indicator

   (4) Initial prospects for risk mitigation

e. Summary of other, non-critical risks

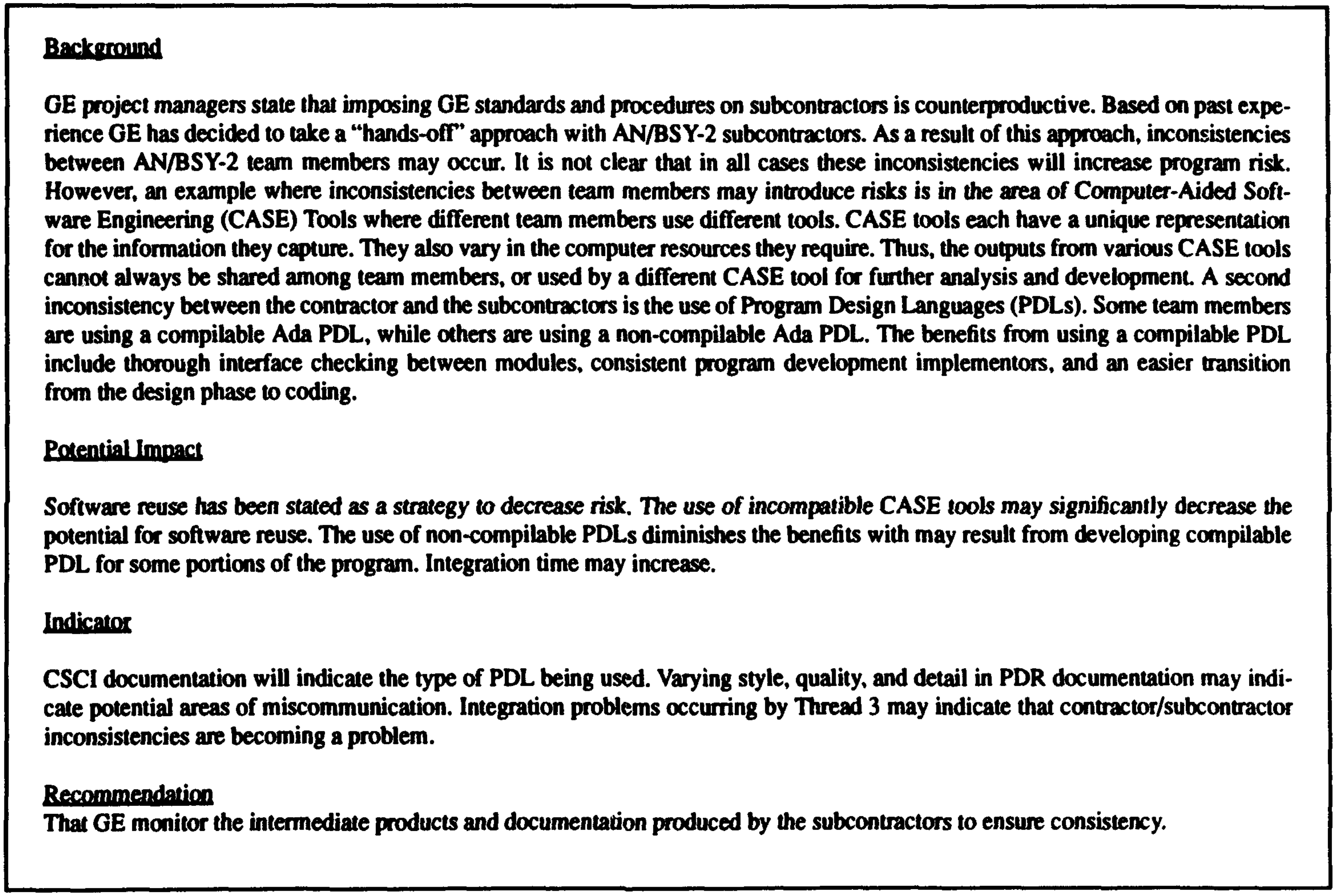
f. Summary of strengths and key achievements of the acquisition

g. Recommendations on risk mitigation planning

Once drafted, the briefing is best coordinated in stages. For example, an initial presentation might be made to working level contractor staff and the program office's software lead. This first run helps fine tune terminology and phrasing for maximum clarity and communication, and exposes weak justifications or contentious conclusions. The next stage might be the presentation to the Program Manager and senior PO staff. The third stage would be a presentation, if required and appropriate, for OSD or senior Service representatives and the PM together.

4.2 DOCUMENTING FINDINGS

The assessment findings, relevant facts, and recommendations must be documented adequately to prevent later misunderstanding should briefing charts only get into wide circulation. It likely will be sufficient to prepare an annotated or scripted version of the briefing for archival and future reference purposes. Annotations or script accompanying the briefing charts provide an opportunity to clarify complex points or issues that arose unexpectedly during verbal presentations. They also serve to record supporting facts that are quickly and concisely verbalized, but lead to tedious briefing charts if fully written down. As an example of this documentation approach, Figures 2 and 3 reproduce one chart and its facing page text from the BSY-2 report.



Background

GE project managers state that imposing GE standards and procedures on subcontractors is counterproductive. Based on past experience GE has decided to take a "hands-off" approach with AN/BSY-2 subcontractors. As a result of this approach, inconsistencies between AN/BSY-2 team members may occur. It is not clear that in all cases these inconsistencies will increase program risk. However, an example where inconsistencies between team members may introduce risks is in the area of Computer-Aided Software Engineering (CASE) Tools where different team members use different tools. CASE tools each have a unique representation for the information they capture. They also vary in the computer resources they require. Thus, the outputs from various CASE tools cannot always be shared among team members, or used by a different CASE tool for further analysis and development. A second inconsistency between the contractor and the subcontractors is the use of Program Design Languages (PDLs). Some team members are using a compilable Ada PDL, while others are using a non-compilable Ada PDL. The benefits from using a compilable PDL include thorough interface checking between modules, consistent program development implementors, and an easier transition from the design phase to coding.

Potential Impact

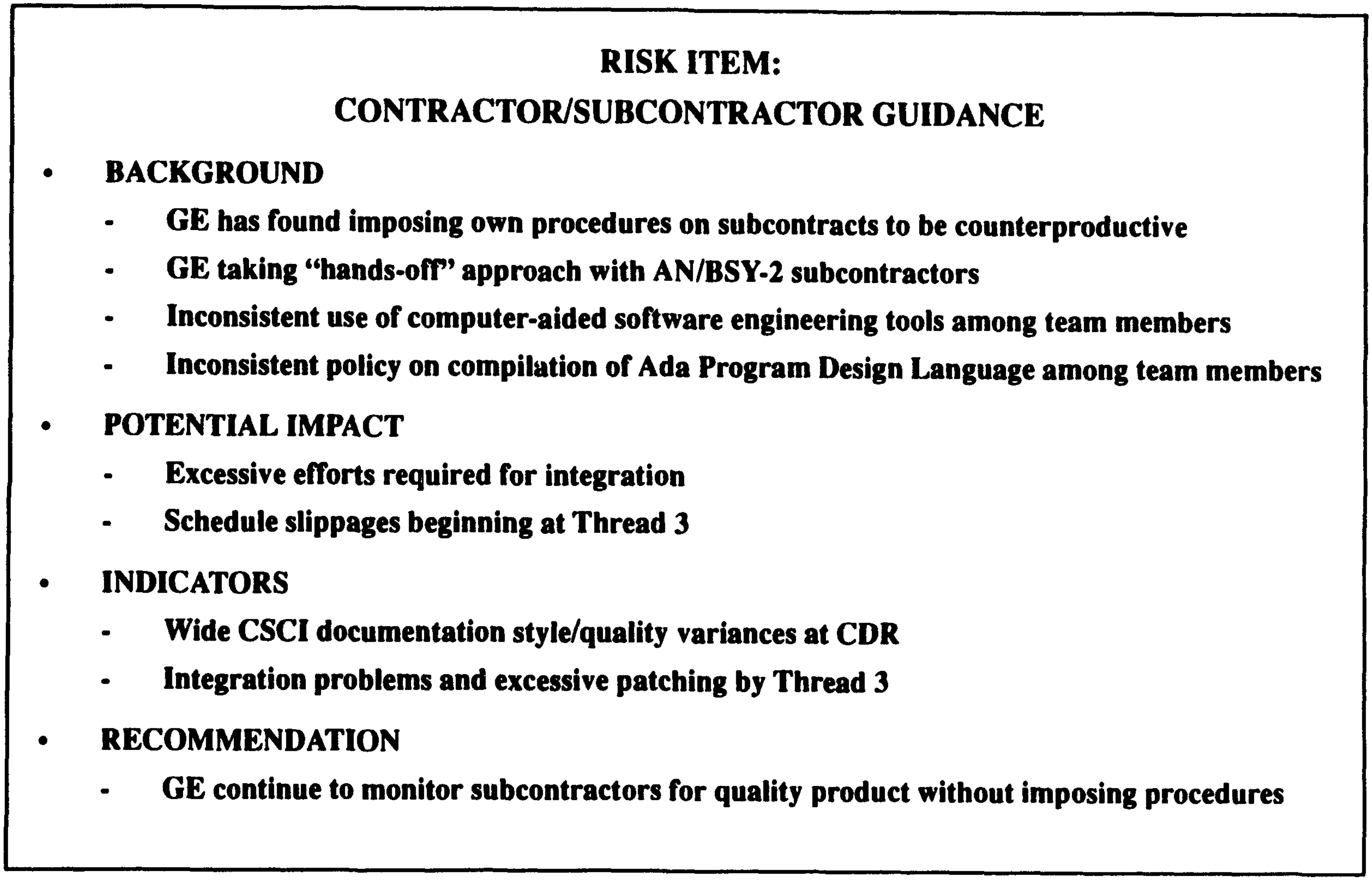
Software reuse has been *stated as a strategy to decrease risk. The use of incompatible* CASE *tools may significantly* decrease the potential for software reuse. The use of non-compilable PDLs diminishes the benefits with may result from developing compilable PDL for some portions of the program. Integration time may increase.

Indicator

CSCI documentation will indicate the type of PDL being used. Varying style, quality, and detail in PDR documentation may indicate potential areas of miscommunication. Integration problems occurring by Thread 3 may indicate that contractor/subcontractor inconsistencies are becoming a problem.

Recommendation
That GE monitor the intermediate products and documentation produced by the subcontractors to ensure consistency.

Figure 2. Example of Facing Page Text from BSY-2 Report



RISK ITEM:

CONTRACTOR/SUBCONTRACTOR GUIDANCE

- **BACKGROUND**
  - **GE has found imposing own procedures on subcontracts to be counterproductive**
  - **GE taking “hands-off” approach with AN/BSY-2 subcontractors**
  - **Inconsistent use of computer-aided software engineering tools among team members**
  - **Inconsistent policy on compilation of Ada Program Design Language among team members**
- **POTENTIAL IMPACT**
  - **Excessive efforts required for integration**
  - **Schedule slippages beginning at Thread 3**
- **INDICATORS**
  - **Wide CSCI documentation style/quality variances at CDR**
  - **Integration problems and excessive patching by Thread 3**
- **RECOMMENDATION**
  - **GE continue to monitor subcontractors for quality product without imposing procedures**

Figure 3. Example of Final Briefing Chart from BSY-2 Report

5. MITIGATING RISK

The last phase of an assessment project is an opportunity to apply an assessment team's expertise to planning solutions to the critical risks and existing problems. The team will have organized a great deal of pertinent data from reviewing the program processes and products, and already will have started to formulate prospective solutions. The goal in this phase is to further develop and evaluate the candidate solutions, while accomodating the Program Manager's stated constraints and guidance.

Risk mitigation effort refines and elaborates proposed solutions, develops cost estimates and initial implementation plans for them, and provides improved analysis of their benefits to the program. The product of the effort addresses each problem or risk found in the basic risk assessment, describes one or more candidate actions or decisions to reduce risk, and evaluates the cost and benefit of each candidate. In formulating solutions, the team should consider whether interrelationships exist among identified risks, and whether large-scale or over-arching risk reduction actions might eliminate many risk sources at once.

Identification of candidate risk mitigating actions leads to three cases. First, as noted already, some risks will be found to be infeasible to mitigate because of prohibitive cost, legal, or other reasons. These risks simply must be accepted. Such cases need to be determined unequivocally, and then set aside so that effort can be concentrated on the remaining risks.

Second, for some risks, risk mitigating actions will be straightforward to implement, with clear benefits well worth their cost. Those cases provide quick and direct benefit to the program, subject to the total cost that is allowable for implementing them.

The third case is difficult. It applies when the benefits from the conceivable risk mitigation actions are as uncertain as the success of the existing program approach. This may mean that a substantial trade analysis or engineering study remains to be done, one that would resolve the uncertainty and shed more light on the program's existing risk. It is better to describe and recommend the needed study than to imply there is no alternative but to accept the existing risk.

Program Managers likely will choose risk mitigation actions with some consideration of factors other than their cost and risk reducing benefits. Perhaps the most useful bottom-line result that a team might offer is a list of recommended actions that is ranked in the order of the ratio of benefit to implementation cost. For those actions where benefit is uncertain and the existing risk is high, the time required to perform needed additional studies also has to be considered in case the results would come too late to be helpful.

Finally, where no reasonable risk mitigation approach is identifiable, some analysis of problem recovery could prove helpful, i.e., what should the PM do when an unmitigated risk materializes to the program's sure detriment? This analysis may identify needed studies that would provide information for recovery decision-making. It also might lead to specific requirements to impose on the current program in the interest of improving recovery opportunities. For example, if a program depends on a particular off-the-shelf software component subject to substantial risk, then some new requirements may be advisable to gain future feasibility of replacing that component with another.

APPENDIX A. SOFTWARE RISK CHECKLIST

This appendix provides a checklist in the form of basic outlines of topics or subjects that may be risk sources for a software acquisition. The checklist's purpose is to help achieve a comprehensive assessment by ensuring that an assessment team considers frequent sources of problems. Use of the checklist requires an assessment team's ability to recognize related subjects and to expand the given information for focused analyses.

The checklist is divided into process and product topics for both contractor and program office. Contractor process topics inc'lude the activities or "key process areas" (KPAs) of the Software Engineering Institute's Capability Maturity Model (CMM)[Paulk 1993]. KPA purpose and goal statements are excerpted largely verbatim. In a few instances, IDA has modified or added to criteria represented in SEI material, to improve specificity or clarity for this guide. It will be beneficial to consult SEI documents for more information and comparisons where important.

IDA has added some contractor process topics to this appendix that are not CMM KPAs. IDA also contributed the program office process topics and the product topics.

Each checklist topic is covered in the same abbreviated way. The description first states the topic's scope and typical risks arising from inadequate methods or results. Then up to five goals related to the topic are stated for an acquisition. If the program under review is achieving these goals, the topic is unlikely to be a major risk source. However, the goals necessarily are general statements until a specific acquisition program is addressed. An assessment team should use the checklist as a starting point to derive more specific criteria and then judge the degree of risk according to the facts of the project under review.

Some product subjects are directly traceable to one or more of the included process topics, e.g., program office Requirements are produced by the program office Requirements Management process. The apparent redundancy is deliberate, to emphasize that the assessment team should obtain and review the content of a documented process result.

A.1 CONTRACTOR PROCESS

A.1.1 Requirements Management

Requirements management establishes common understanding between customer and developer about the customer's requirements that will be addressed by the project.

Risks from inadequate requirements management by a contractor include incomplete or ambiguous requirements, insufficient disclosure of derived requirements to the customer, and failure to provide a thorough requirements baseline to control ongoing development work.

Primary goals are

a. System requirements allocated to software are controlled to establish a baseline for software engineering and management.

b. A thorough approach is taken for deriving and documenting software requirements, including a specification method supported by a Computer-Aided Software Engineering (CASE) requirements analysis and design tool.

c. Software requirements are elaborated to exhibit all important operational capabilities to the customer and end user.

d. As software requirements are elaborated into design specifications, traceability and justifiability relative to customer-provided requirements are documented.

e. Software plans, products, and activities are kept consistent with the allocated software requirements.

A.1.2 Software Project Planning

Software project planning establishes reasonable plans for performing the software engineering and managing the software project.

Risks from inadequate project planning include omission of essential activities, inappropriate commitments, and failure to update plans as commitments change.

Primary goals are

a. Software estimates are documented for use in planning and tracking the software project.

b. Software project activities and commitments are planned and documented (e.g., in a Software Development Plan per DoD-2167A).

c. Affected groups and individuals agree to their commitments related to the software project.

d. Software estimates are derived from a defined process based on historical and analogous results and are supported by tools and experienced personnel.

e. Software schedules and milestones are derived from an estimation process and updated regularly to reflect changes in requirements or commitments.

A.1.3 Software Project Tracking and Oversight

Software project tracking and oversight establishes adequate visibility into actual progress so that contractor management can take effective action when the software project's performance deviates significantly from plans.

Risks from inadequate project tracking and oversight include cost overrun and missed milestones for individual tasks, and authorizing development work that is inconsistent with project risks and priorities.

Primary goals are

a. Actual results and performance are tracked against the software plans.

b. Corrective actions are taken and managed to closure when actual results and performance deviate significantly from plans.

c. Changes to software commitments are agreed to by the affected groups and individuals.

A.1.4 Software Subcontract Management

Software subcontract management aims to select qualified software subcontractors and to manage them effectively.

Risks from inadequate performance include inadequate communication and understanding of system and software requirements and of program commitments.

Primary goals are

a. The prime contractor has selected qualified software subcontractors.

b. The prime contractor and the software subcontractor understand and agree to their commitments to each other.

c. The prime contractor and the software subcontractor maintain ongoing communications.

d. The prime contractor tracks the software subcontractor's actual results and performance against commitments.

A.1.5 Software Quality Assurance

Software quality assurance provides management with appropriate visibility into the process being used by the software project and of the products being built.

Risks from inadequate quality assurance effort include undetected or unrecognized defects, and quality assurance findings that are not resolved promptly and completely.

Primary goals are

a. Software quality assurance activities are planned.

b. Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.

c. Affected groups and individuals are informed of software quality assurance activities and results.

d. Noncompliance issues that cannot be resolved within the software project are addressed by senior management.

A.1.6 Software Configuration Management

Software configuration management establishes and maintains the integrity of a software project's products throughout the project's life.

Significant risks from inadequate performance include inadequate control of different product versions, unapproved revision of baselined products, and inability to properly trace product revisions to problem reports and approved change proposals.

Primary goals are

a. Software configuration management activities are planned.

b. Selected software work products are identified, controlled, and available.

c. Changes to identified work products are controlled.

d. Affected groups and individuals are informed of the status and content of software baselines.

A.1.7 Organization Process Focus

Organization process focus establishes a contractor organization's responsibility for software process activities that improve the organization's overall software process capability.

The major risks from inadequate performance are that a project's development approach and infrastructure are largely self-created and maintained, rather than derived from proven corporate experience.

Primary goals are

a. Software process development and improvement activities are coordinated across the organization.

b. The strengths and weaknesses of the software processes used are identified relative to a process standard.

c. Organization-wide process development and improvement activities are planned.

A.1.8 Organization Process Definition

Organization process definition develops and maintains a usable set of software process assets for the contractor organization, in order to improve performance across all software projects and provide a basis for cumulative, long-term benefits to the organization. Process assets include policies, technical guidelines, tools, and experience data.

The major risk from inadequate performance is that a project's process assets are largely project-unique and thus possibly less mature, less familiar, less complete, and more costly than organization-wide assets may be.

Primary goals are

a. A standard software process for the organization is developed and maintained.

b. Information related to the use of the organization's standard process by projects is collected, reviewed, and made available.

A.1.9 Training Program

A *training program* aims to develop the skills and knowledge of contractor staff so that they can perform their assignments effectively and so that each project has planned training as a means to fulfill its staffing needs.

A major risk is that software designers and implementers with required skills and experience are not readily available for the project.

Primary goals are

a. Primary skill needs for the program under review are identified and described for planning and scheduling training.

b. Training is being accomplished to develop the skills and knowledge needed for performing software management and technical roles on the project.

c. Assignments and work experiences for new staff are planned and conducted to support training objectives.

A.1.10 Integrated Software Management

Integrated software management integrates software engineering and management activities into a coherent, defined software process that is tailored from the contractor organization's standard software process and related process assets.

Risks of an inadequately defined or incomplete process are omission of project activities required by the customer, and inconsistent handling of similar work products in progress.

Primary goals are

a. The project's defined software process is a tailored version of the organization's standard software process.

b. The project is planned and managed according to the project's defined software process.

A.1.11 Software Product Engineering

Software product engineering performs a well-defined engineering process that integrates all the software engineering activities to produce correct, consistent software products effectively and efficiently.

Risks related to inadequate product engineering include failure to meet all requirements for a deliverable product and failure to control resources expended on tasks in proportion to their value for each deliverable.

Primary goals are

a. The software engineering tasks are defined, integrated, and consistently performed to produce the software.

b. Software work products are kept consistent with each other.

A.1.12 Intergroup Coordination

Intergroup coordination establishes a means for the software engineering project group to participate actively with other contractor engineering groups so that the project is able better to satisfy the customer's needs effectively and efficiently.

Risks related to inadequate coordination include omitted or unauthentic software requirements and attendant late discovery of serious defects.

Primary goals are

a. The customer's requirements are agreed to by all affected groups.

b. The commitments between the engineering groups are agreed to by the affected groups.

c. The engineering groups identify, track, and resolve intergroup issues.

A.1.13 Peer Reviews

Peer reviews aim to remove defects from software work products early and efficiently.

Potential risks related to inadequate or omitted reviews include a high level of defects and very costly rework to remove them in the late stages of software development, e.g., integration testing.

Primary goals are

a. Peer reviews are planned.

b. Defects in the software work products are identified and removed.

A.1.14 Quantitative Process Management

Quantitative process management controls the process performance of the software project quantitatively.

The risk associated with lack of quantitative process management is inability to relate process performance (e.g., defects detected) to individual process activities and use of process assets.

Primary goals are

a. The quantitative process management activities are planned.

b. The process performance of the project's defined software process is controlled quantitatively.

c. The process capability of the organization's standard software process is known in quantitative terms.

A.1.15 Software Quality Management

Software quality management develops quantitative understanding of the quality of a project's software products and achieves specific quality goals.

The risk associated with inadequate quality management is inability to quantify and control product quality.

Primary goals are

a. The project's software quality management activities are planned.

b. Measurable goals for software product quality and their priorities are defined.

c. Actual progress toward achieving the quality goals for the software products is quantified and managed.

A.1.16 Defect Prevention

Defect prevention identifies the cause of defects and prevents them from recurring.

The major risk related to inadequate defect prevention is a high level of defects occurring throughout a project from the same or similar causes.

Primary goals are

a. Defect prevention activities are planned.

b. Common causes of defects are sought out and identified.

c. Common causes of defects are prioritized and systematically eliminated.

A.1.17 Technology Change Management

Technology change management identifies new software technologies, both for the contractor organization's deliverable products and for its development process, and tracks them into the organization in an orderly manner. Examples of new technologies that an assessment team might encounter include object-oriented analysis and design, parallel or distributed processing, artificial intelligence, database machines, computer speech recognition, and neural net technology.

Significant risks from inadequate technology change management are poor quality in software releases incorporating new technologies, and inadequate understanding of the beneficial applications of a technology.

The primary goals are

a. Incorporation of technology changes is planned, e.g., technology assessments evaluate each new technology and plan the activities and support needed for exploiting it successfully.

b. New technologies are evaluated to determine their effect on quality and productivity.

c. Appropriate new technologies are transferred into normal practice across the contractor organization and within the project.

A.1.18 Process Change Management

Process change management continually improves the software processes used in the contractor organization with the intent of improving software quality, increasing productivity, and decreasing product delivery time.

The major risk related to inadequate process change is inability to sustain and improve software quality and cost factors.

Primary goals are

a. Continuous process improvement is planned.

b. Participation in the organization's software process improvement activities is organization wide.

c. The organization's standard software process and the projects' defined software processes are improved continuously.

A.1.19 Software Trust Management

Software trust management extends the contractor's software development process and process assets, in order to prevent maliciously introduced software vulnerabilities, security weaknesses, and development efforts in conflict with project needs.

Because software is so prone to design flaws, malicious effort disguised as natural human mistakes is a risk. Also, extraneous features and code, though undertaken for legitimate reasons, may increase perceived defects and be exploited for improper purposes.

Examples of a software trust methodology are from the Ballistic Missile Defense program [Watson 1992] and the computer safety fields [IEEE 1992].

Primary goals for software trust management are

a. Software trust is a recognized goal and trust management activities are planned throughout the contractor's process.

b. There is a high level of shared knowledge within the development team, provided by means such as buddy roles, rotating assignments, peer reviews, and inspections.

c. All development activity is traceable to requirements, and personal responsibility for product results is evident.

d. Process assets and work products are protected against inappropriate access, unauthorized change, and accidental loss.

A.1.20 Performance Management

Performance management establishes the process and product foundations for meeting and exceeding required performance and computer resource utilization targets, insofar as feasible with the host or target system capabilities.

The major risk of inadequate performance management is late discovery that required performance or resource utilization targets (e.g., reserve memory capacity) cannot be achieved by the software as designed on the selected host system. Typically this leads to major software rework, cost overruns, and missed delivery.

Primary goals are

a. Performance and computer resource utilization requirements are allocated to key software components and operations in the derived software requirements.

b. Development standards and training provide guidance for design and implementation to meet performance goals.

c. Performance and computer resource utilization are measured throughout development and testing.

d. Performance risks are identified and tracked throughout the software life cycle, and recognized in planning software rework or host system upgrade.

A.1.21 Risk Management

Risk management establishes actions and priorities for reducing the impact of negative events or conditions on which software project success depends.

Without risk management effort, contractor decision-making may lack consistent criteria and may not prepare the project to deal adequately with foreseeable risks.

Primary goals of risk management are

a. Significant project risks are identified and assessed periodically and as events may warrant.

b. Project planning considers feasible actions to mitigate risks.

c. Risk assessment and mitigation alternatives are communicated to the customer.

d. Risk mitigation actions are implemented.

A.2 PROGRAM OFFICE PROCESS

A.2.1 Acquisition Planning

Acquisition planning establishes the procurement approach and basic program constraints for acquiring the envisioned software, including necessary software engineering, development, and integration.

A range of potentially critical risks pertain because of the singular importance of this activity. Examples include failure to recognize and mitigate overall program risks, unreasonable budget and schedule targets, inadequate source competition, and inadequately stated requirements.

a. Acquisition planning is based on authentic requirements and risk assessment.

b. Competition is open to the most technically qualified sources.

c. Software risk is mitigated by encouraging maximum use of commercial standards and software products, and by prototyping.

d. Incremental and evolutionary development is planned.

e. Planned contractor roles and interactions are practical.

A.2.2 Requirements Management

Requirements management by an acquisition program office establishes mutual understanding with contractors and operational end users of the capabilities that must be delivered in the system being developed, and also controls baseline requirements as a critical factor in successful program management and system delivery.

Major risks from inadequate requirements management are end user rejection of contracted requirements and failure to sustain contractor focus on critical operational needs.

Primary goals are

a. Operational end users are clearly identified and participate in requirements management.

b. Technical interchanges are conducted to help prospective contractors understand the requirements and make recommendations before competition ensues.

c. An operational concept description for the deliverable system is approved by end users before preliminary design is completed.

d. Government requirements are progressively refined and elaborated by contractors, leading to preliminary design.

e. Requirements are evaluated relative tc precedent systems and contractor experience, in order to understand risks.

f. Informative and rigorous traceability is maintained throughout the program among government-stated requirements and contractor work products.

A.2.3 Project Tracking and Oversight

Project tracking and oversight identifies the software project status and trends relative to scheduled tasks, resources, and expenditures.

Inadequate project tracking and oversight lead to risk of cost overrun and schedule slip, and inability to document a project's evolution.

The primary goals are

a. Contractor plans and schedules are assessed to know their basis and risks.

b. Estimates of software development schedule and costs are made independently of contractor estimates.

c. Technical and development status of major products, e.g., Software Requirements Specification or source code for each Computer Software Configuration Item, are reviewed with contractors monthly and issues are tracked to closure.

d. Cost, schedule, problem reports, and other metrics are reviewed for consistency with development status and for projecting near-term trends.

e. The impact of prospective program decisions is assessed fully before commitment.

A.2.4 Staffing and Training

Staffing and training address program office needs for personnel with sufficient software skills to develop requirements, assess software technology and lessons learned, review contractor work, and provide technical program guidance.

Risks related to inadequate program office staffing include an inadequate or nonobjective basis of government requirements and technical direction to contractors.

a. A program office software lead is designated who has direct software development experience.

b. Software staff receive training in modern software technology and development practices appropriate for the acquisition.

c. The number of software knowledgeable staff in the program office is consistent with anticipated software cost and risk relative to the overall acquisition.

A.2.5 Validation and Verification

Validation and verification (V&V) improves the program office's technical visibility of contractors' software work products and helps assure that the delivered system will be operationally satisfactory to intended users.

Major risks of inadequate or omitted V&V include a high level of software defects, unrecognized departures from approved requirements, and acceptance of technically unfounded engineering choices.

Primary goals are

a. Validation and verification activities are planned and performed by well-qualified groups or individuals who are independent of the contractors' project staff.

b. Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively.

c. Affected groups and individuals are informed of V&V activities and results.

d. Noncompliance issues are addressed and resolved promptly by appropriate contractor management or government contracting officials.

A.2.6 Configuration Management

Configuration management controls the requirement and product baselines for the program, their interrelationships, and the process for making changes.

Potential risks related to inadequate configuration management include implicit acceptance of unnecessary work, unrecognized rippling of changes onto multiple products, and inability to relate products and requirements.

Primary goals are

a. Requirements baselines and traceability are maintained independently of the development contractors.

b. A widely understood, disciplined procedure is used to request, evaluate, and dispose of changes to government requirements and contractor work products

c. Products critical to future program management and deployed product support are kept consistent with each other and with the applicable requirements baseline.

A.2.7 Contractor Teams Coordination

Contractor teams coordination refers to program office responsibilities to coordinate planning and problem solving among mutually supporting, but independent teams. This need may arise from distinct but coordinated acquisition programs or within one program in which different elements of a "system of systems" are separately procured.

Risks from inadequate coordination include program schedule or cost risks when one team's problems or plans are not fully known and accommodated by other teams.

Primary goals are

a. Events and milestones requiring coordination are planned and accepted by all affected teams and responsible program office staff.

b. Interface requirements are thoroughly specified, independently verified, and accepted by affected teams with sufficient lead time.

c. Program office staff responsible for different teams participate in all contractor reviews.

d. Technical interchanges are conducted between interfacing contractor teams throughout development, with documented conclusions or findings.

A.2.8 Reviews

Reviews address software status and technical issues in an open, peer-driven assessment, to make all risks evident for resolution by program office and contractor management.

Risks of inadequate reviews include insufficient program office understanding of status and impacts, and failure to address high priority project needs.

Primary goals are

a. Reviews are planned and conducted to expose the progress, problems, and views of individual contractors involved in software development and integration.

b. Documents for review are thorough and provided well in advance of review meetings.

c. Peer reviewers are technically qualified and independent of developers.

d. End users consistently participate in program reviews.

e. Issues are adequately defined for action and tracked to closure.

A.2.9 Risk Management

Risk management establishes actions and priorities for reducing the impact of negative events or conditions regarding software on which program success depends.

Without risk management effort, program office decisions may lack consistent criteria and may not deal adequately with foreseeable risks.

Primary goals of risk management are

a. Significant program risks affecting software development are identified and assessed periodically and as events may warrant.

b. Risk assessments are reviewed by independent experts.

c. Risk mitigation is planned and timely recommendations communicated to the program manager and responsible acquisition executive.

d. Risk mitigation actions are implemented.

e. The program manager and acquisition executive recognize and accept risks that cannot be mitigated economically.

A.2.10 Software Quality and Trust Management

Software quality and trust management determines contractors' achieved software quality and trust, and advises the contractors and program manager of recommended actions for improvement.

Risks from inadequate performance are lack of evidence that quality and trust meet defined requirements or reasonable standards of practice.

Primary goals are

a. Program office quality and trust assurance activities are planned.

b. Software quality and trust objectives are defined quantitatively and objectively.

c. Contractor quality assurance and trust evaluation efforts are continually monitored to know status and results throughout the project.

d. Contractors are informed of unacceptable and undesirable trends, and recommended corrective actions.

A.2.11 Test Planning

Test planning establishes the criteria and approach for confirming usability and conformance to requirements before deployment of the deliverable software.

Inadequate test planning presents risks of accepting unsatisfactory software that negatively impacts end user operations and delivery delays due to unavailability of detailed test plans and resources.

Primary goals of test planning are

a. Test planning activities are planned within the overall program approach.

b. Required testing activities, responsibilities, and resources are defined with sufficient lead time.

c. Acceptable testing criteria are established and managed as part of requirements.

d. End users contribute to and approve test plans.

e. Planned testing is accomplished.

A.2.12 Deployment Planning

Deployment planning establishes the approach and requirements for transitioning the delivered system and software to end users. Issues addressed may include user training, beta site testing, special tools or operational artifacts needing development, etc.

Risks with inadequate deployment planning include omitted development requirements, e.g., training resources, and inadequate lead time to meet a given deployment date.

Primary goals are

a. Deployment requirements are defined with sufficient lead time.

b. Deployment needs are addressed in developmental requirements.

c. End users contribute to and approve deployment plans and requirements.

d. Deployment readiness is achieved before release and deployment of software

A.3 CONTRACTOR PRODUCTS

Product topics concern the technical content of products forthcoming from a software project, in contrast to process topics, which address how the results are produced (i.e., work practices). The following topics are meant to focus on technical content issues rather than standard document types that may be used for delivering the information. In some cases, the information may not be a required deliverable to the acquisition program office.

A.3.1 Software Development Methodology

Software development methodology defines the contractor's technical process for producing software that will meet the government-stated requirements. This is an integrated view of engineering tasks, methods and analyses, tools, and decision criteria that will apply in performing the needed software engineering and deciding that products have been successfully completed. Typically this is provided in a Software Development Plan, with appropriate updates as the project progresses, and is a result of the contractor's Integrated Software Management process (see A.1.10).

Risks of inadequately defined methodology include ad hoc or inconsistent design practices and loss of technical control over partial products, their quality, and interrelationships.

Primary goals for software development methodology and tools are

a. All needed software engineering and development tasks are identified and their objectives, methods, analyses, tools, and engineering criteria are defined.

b. Development methodology is directly related to identified program risks and major challenges.

c. Methods and measurements for ensuring project conformance to the methodology are defined.

d. Development methodology is traceable to contractor experience and standardized organizational practice.

A.3.2 Software Development Tools and Environment

Software development tools and environment refer to the specific toolset and associated interoperability mechanisms that form the automated environment supporting the project's software development process.

Risks with an inadequate environment include limited and immature tools, difficulty in reusing products from one tool to another, and lack of tools to support key activities.

Primary goals for the selected tools and environment are

a. The toolset provides robust support for the contractor's process and for software development on the application's target platform.

b. The toolset and interoperability mechanisms are predominantly COTS items.

c. Compelling reasons exist for use of any contractor-proprietary tools, and such tools have no long-term impact on supportability of the deliverable software.

d. The toolset and environment is well based on prior contractor experience and requires minimum installation, set-up, and shakedown before project use.

A.3.3 Development Standards and Guidance

Development standards and guidance provide individual designers and programmers with the most pertinent technical guidance for implementing a quality, well performing product.

Risks from inadequate standards are arbitrary or uninformed design and implementation choices and inconsistent implementation practice across the Computer Software Components (CSCs) of the application system.

Primary goals for development standards and guidance are

a. All areas where individuals may need guidance are addressed, if only by reference to well known technical references.

b. Critical issues for the system under development are described and specific guidance is provided for acceptably resolving the issues within this project.

c. Guidance is supported by cited contractor experience, trade analyses, or benchmark measurements.

d. Tools, inspections, and coaching are provided to achieve conformance.

A.3.4 Derived Software Requirements

Derived software requirements are produced early in contractor effort to understand government-stated requirements and develop a preliminary software architecture and design.

The major risk from inadequate contractor-derived requirements is an incomplete or incorrect basis for controlling further design work and project scope.

The primary goals for derived software requirements are

a. Requirements are individually related to operational functions and goals.

b. Operational features not explicitly identified by the government are defined and proposed for inclusion in system design.

c. Completeness and consistency of government-stated requirements is either confirmed or else necessary requirement changes are proposed.

d. The principal design and implementation challenges are identified and analyzed, and applicable precedent systems and standards are identified.

A.3.5 Prototyping Plan

Prototyping plan addresses how uncertain requirements and primary technical challenges will be investigated to resolve risks and define acceptable design concepts.

The risks from an inadequate plan are lack of needed evaluation data and late discovery of misunderstood requirements or unacceptable design.

Primary goals for a prototyping plan are

a. Available experience and precedent systems are assessed to define the scope of required prototyping and the design or requirements issues to be resolved.

b. Prototyping issues are separated and addressed incrementally by successive prototyping efforts.

c. Prototyping methods are consistent with the scope and importance of the issues to be resolved, and objective decision criteria are stated for resolving issues.

d. Prototype development plans are consistent with project schedule and goals for reusing prototype software in the deliverable system.

A.3.6 Software Architecture and Design

Software architecture and design is produced from preliminary design effort. It identifies the configuration of components for the deliverable system, describes their operation and interactions, and outlines design concepts and standards for important system functions such as user interface, data management, mission algorithms, distributed processing.

Risks from an inadequate software architecture and design include inability to determine how well requirements will be met and inability to gain end user acceptance of the design approach.

Primary goals for software architecture and design are

a. The architecture and design communicates how well the most significant government requirements will be met.

b. Techniques and concepts to meet operational needs and derived requirements are elaborated.

c. Reused and COTS software components are identified fully and justified by analyzing their capabilities and integration requirements.

d. Trades and alternatives are analyzed to justify design choices.

A.3.7 Host System Selection

Host system selection establishes the choices and configuration of computing, peripheral, and networking devices and general-purpose software, e.g., operating system and LAN software, that are the execution platforms for the software being developed. To ensure satisfactory performance and operational service, the software developer must have significant responsibility in host system selection.

The major risks from an inadequate host system are inadequate operational performance or responsiveness, limited availability of commercial software components for use in the application, and limited options for upgrading component capabilities.

The goals for host system evaluation and selection are

a. Host system selection is driven by software requirements.

b. Practical alternatives are sought and compared objectively.

c. Except for compelling military necessity, the chosen host system components are commercial, off-the-shelf (COTS) products.

d. A performance and capacity model for the overall system is used to determine that each component is adequate and has a reasonable margin of capability above the maximum anticipated demand.

e. Host components with major influence over system performance are upgradable to higher performance or capability without modifying application software.

A.3.8 Test Plan & Resources

Test plan and resources establishes the capability to adequately confirm usability and conformance to requirements before delivery of the software

Risks from inadequate planning for testing and test resources include incomplete testing and quality assurance effort throughout the project, and inability to meet project testing schedule for lack of resources such as host system and test data.

Primary goals for test plan and resources are

a. Testing and related quality assurance methods (e.g., inspections) are defined and planned throughout the project.

b. Needed test resources and responsibilities for providing them are specified.

c. Test plan meets or exceeds government-stated objectives and criteria.

d. Test plan and resources provide an effective approach to both defect detection and confirmation of operational suitability.

A.3.9 Staffing Plan

Staffing plan identifies on-board project personnel and planned hires relative to project schedule, roles, activities, and necessary skills.

The major risks of inadequate staff planning include failure to adequately identify personnel shortfalls and failure to control labor expenditures according to project priorities and risks.

The goals for a staffing plan are

a. Pertinent skill and labor requirements are identified relative to scheduled tasks.

b. Future hires are planned according to needed skills and task requirements.

c. Program impacts due to personnel changes and training are traceable.

A.3.10 Source Code

Source code is the computer instructions of the deliverable software as written in a high order language such as Ada. An assessment team typically would do no more than simple reviews of source code samples for the purpose of confirming standards and practices.

The major risk in inadequate source code is lack of maintainability due to nonconformance to established standards, poor structure and design, and insufficient in-line comments.

The goals for source code are

a. Status of the source code is consistent with status as reported in configuration and project management records, e.g., unit development folders.

b. Source code conforms to pertinent project standards.

c. Source code functionality is consistent with detailed requirements and traceability information.

A.4 PROGRAM OFFICE PRODUCTS

Product topics for an acquisition program office address desired content of the more important software work products, independently of specific application or operational features of the system being developed. The following concentrates on technical content issues rather than the official documents that may convey the information.

A.4.1 Requirements

Requirements state

1. necessary work activities, usually in a Statement of Work (SOW),
2. deliverable work products, via a Contract Data Requirements List (CDRL),
3. performance, functions, and other characteristics needed in the deliverable system, in a Technical Requirements Document (TRD) or other specification (e.g., an SSS).

Software requirements may not be explicit or separately called out from computer and other hardware requirements when the acquisition covers a complete system.

Major risks from unsatisfactory requirements include an inadequate basis for deriving software requirements, controlling the software project, and delivering a usable system.

Primary goals for program office requirements, stated at the system or overall program level, are

a. Overall system requirements identify where software applies and where COTS and reused products are required or desirable.

b. The form and notations for stating requirements assists review for completeness and consistency.

c. Practical ways to test satisfaction of requirements are evident.

d. Requirements specify needed operational capabilities and performance for the system, and impose design constraints or solutions only where essential for deployment, interoperability with other systems, and system evolution.

e. Unquantifiable system requirements are covered by contractor activities identified in the SOW, with objectively measurable criteria specified for their scope and satisfactory completion.

A.4.2 Risk Management Plan

Risk management plan evaluates foreseeable risks affecting software performance, cost, delivery schedule, and supportability, and describes and costs actions to reduce risk impact, making recommendations for action to the program manager.

Risks with an inadequate risk assessment and mitigation plan are failure to recognize substantial risks and failure to provide alternatives to cope with negative impacts.

Primary goals for the risk assessment and mitigation plan are

a. Risk identification is comprehensive and independently verified.

b. Statements of potential impact are thorough and realistically stated in terms of quantitative effects on program goals and milestones.

c. Mitigation alternatives adequately address the potential impact of each risk.

d. Recommended actions are ranked by risk mitigation benefit and cost.

A.4.3 Government Furnished Equipment & Software Selection

Government-furnished equipment (GFE) and software refers to items that will be integral parts of the deliverable system. Thus their *selection* defines constraints on the application system design in the interest of economy, interoperability, or reuse.

Major risks that may be introduced by the selected GFE or software include inadequate components, added contractor cost to gain working knowledge of the furnished items, and schedule and cost risks related to delivery and support of the furnished items.

Primary goals for the selection are

a. The furnished items have limited and isolated effect on the application software architecture and design as well as the software development effort.

b. Available support for the furnished items is comparable to available support for COTS products of similar capability, or is readily made so.

c. Substantial technical or interoperability benefits accrue from using the selected items, besides economic benefits.

A.4.4 Test & Integration Resources

Test and integration resources are the government-furnished capabilities to help integrate different software and hardware components into a working system, and adequately confirm its usability and conformance to requirements. The resources may include personnel, test subjects, test harnesses, special operational data, etc.

Risks from inadequate test and integration resources include delay in undertaking final system testing and incomplete testing.

Primary goals for test and integration resources are

a. Test and integration resources are available when needed.

b. Test and integration resources are mature or proven in practice, and operationally well supported.

c. Test and integration resources support an effective level of acceptance testing.

A.4.5 Development Status Data

Development status data is the PO's basis for understanding and tracking the status of the evolving deliverable software and gaining quantitative insight in advance of development problems.

The risks of inadequate status data include inability to forecast the remaining work schedule and inability to foresee significant problems in delivering software on schedule.

Goals for development status data include

a. Status is tracked throughout the development at a level consistent with the major work activities in progress, e.g., at CSCI level during preliminary design but at the more detailed level of individual software units or programs during detailed design, code inspections, and testing.

b. Status information is derived from both contractor data and independent observations.

c. Status is measured by several metrics including estimated size of code, defined work milestones accomplished, reported defects.

d. Status is aggregrated to provide an overall measure of product status.

APPENDIX B. THE IDA BSY-2 ASSESSMENT PROCESS

In the fall of 1989, IDA conducted a software assessment of the AN/BSY-2 Submarine Combat System. The study was sponsored by the Director, Naval Warfare and Mobility in the Office of the Deputy Director of Defense for Acquisition, Tactical Warfare Programs (TWP). The following briefly describes the overall process evolved by the IDA team for the assessment. A complete report of the study and its findings is [Donis 1990].

a. The assessment team obtained an in-house overview of the SSN-21 system, its capabilities, and the role that BSY-2 will play for that submarine.

b. The task leader requested the Navy PO to provide various documents. Mainly, these were 2167 documents such as the System Requirements Specification, System Design Document, CSCI design documents, the Software Development Plan, etc. In addition, the Test and Evaluation Master Plan, the Master Test and Evaluation Plan (the contractors version of the TEMP), the Naval Underwater Systems Center (NUSC) IV&V plan, and other documents were requested.

c. The assessment team spent about two weeks reading the documentation. From these readings, the team became familiar with the software development approaches used by the contractor, General Electric (GE). For example, the team examined:

   (1) The design approach for all CSCI's

   (2) The design for specific CSCI's

   (3) The tools used for design and development

   (4) The IV&V approach

   (5) The data management approach

   Members of the assessment team were assigned various areas of technical responsibility, most of which later became risk categories.

d. After digesting the available information, the team developed a list of comments and questions that surfaced from assigned readings. The list was provided to the PO and feedback was requested.

From these documents, the team able was to identify the potential risk items that were judged to require attention. For example:

(1) Using a COTS relational database management system (Ingres) for real-time applications was unusual. It was unlikely that Ingres would perform properly in a real-time environment. The team was interested in finding out more about this data management approach.

(2) At the time, GE did not have extensive expertise in designing with Ada. For example, the GE standards and practices manual did not provide necessary guidance for using key Ada features (e.g., tasking and rendezvous) or resource management. Inadequate expertise and training in Ada could spell disaster at a later date.

(3) In looking at the GE design for the network topology, GE specified that 2000 messages per second were required for system functions. Based on current practice for similar communications networks, this rate seemed unrealistic.

Potential risk items were placed into the following categories:

(1) Programming language

(2) Enhanced Modular Signal Processor (EMSP)

(3) Testing

(4) Network design

(5) Data management

(6) Development management

(7) Use of standards

(8) Operator interface

Approximately three-fourths of the eventual critical risks were first noticed during the documentation review.

e. The next step involved site visits. The whole team visited NUSC to learn about its role as technical direction agent. The team then went to GE in Syracuse to ask detailed questions about the potential risks that had been identified. The team was able to interact with the system designers, including those adapting the EMSP, the data management group, the Ada educator, etc. The team visited the EMSP facilities, and a member of the team was able to converse one-on-one with the GE EMSP expert. The team viewed the prototype screens, the computer-assisted software engineering (CASE) tools, and other artifacts.

As a result of the information received from the site visit, the team was able to confirm the majority of the potential critical risks found prior to the visit.

New risk areas were also identified by the contractor during the site visit, and the contractor's risk management approach was discussed.

f. The team then reexamined all risk areas and discussed their potential impact. Each risk area was deemed critical or non-critical. Critical risks areas were those with high likelihood of becoming problems that would either delay delivery schedule or result in failure to satisfy mission requirements. The team then defined indicators or criteria that could be monitored in the future to give an early signal that each risk was developing into a problem.

g. The team briefed the Navy PO and TWP on the preliminary results.

h. A final report was then produced.

APPENDIX C. SOFTWARE CAPABILITY EVALUATION EXPERIENCE

In 1991, IDA began providing technical advice and support to the Ballistic Missile Defense Organization (BMDO) on the use of the Software Engineering Institute's (SEI's) Software Capability Evaluations (SCEs). IDA helped BMDO develop software policy which recommends the use of SCEs for both source selection and contract monitoring. As of Summer 1993, the Brilliant Eyes (BE) program has used SCEs for source selection. The National Test Facility (NTF) and the Brilliant Pebbles (BP) programs have used SCEs to help monitor their contractor's software activities after contract award. Other BMDO programs are scheduled to use SCEs in the near future; e.g., GBI and BM/C3.

Over the last three years, IDA has been quite involved in institutionalizing SCEs across BMDO. We helped to inform BMDO of the SCE process and potential benefits, to organize the evaluations, and to coordinate SCE schedules and training courses. As a result, IDA has become directly involved supporting BMDO elements, understanding contractors software development weaknesses, and improving the SCE process. The following briefly describes the overall SCE process and the experience of the BE, BP, and NTF evaluation teams. Further information is found in [Springsteen 1991] and [Springsteen 1992].

C.1 PREPARING FOR ASSESSMENT

C.1.1 Plan and staff project

Several things must be done to properly plan and staff an SCE.

Inform the program office. Substantial lead time is required to successfully implement SCEs into the source selection process (approximately 6 months.) The element program managers and software leads first must be briefed on what SCEs are, how they are performed, and how the results are used in source selection. The program management must be convinced that SCEs are worthwhile before they sign up to them. Substantial time also may be spent informing legal and contracting personnel about the nature of SCEs and persuading them that SCEs are not an unprecedented activity.

Develop inputs for source selection process. The contractor community must be notified in advance that an SCE will be performed. For both the BE and BP programs, the Statement of Work (SOW) and Request for Proposal (RFP) contained statements indicating a government team would perform an SCE and provide the results to the program office. It is beneficial to have prepared sections for the SOW and RFPs so that program office representatives can insert the proper paragraphs into their documents without having to spend time writing them.

Establish evaluation criteria. To use the SCE results in the BE source selection process, evaluation criteria were defined during the RFP preparation phase. The evaluation criteria defined how the SCE results would be used to assign a color rating (i.e., red, yellow, green, blue). SEI does not provide guidance for establishing the evaluation criteria. It was unclear how much detail to include in the evaluation criteria and how much summarizing the SCE team should perform to best support the Source Selection Evaluation Board (SSEB).

Select the evaluation team. The composition of the BE and BP evaluation teams were derived based on obtaining a cross section of military service expertise in software. Both the BE and BP SCE teams consisted of members from the Army, Air Force, Federally Funded Research and Development Centers (FFRDCs), and National Laboratories. The FFRDC members of the team helped to provide software development expertise and the government program office members provided acquisition knowledge. It is important to have at least two or three team members with strong software development experience as opposed to software acquisition experience.

Train the evaluation team. SEI requires that the SCE team members attend a 4 day training course in Pittsburgh, PA. You must allow approximately 3-4 months to register the team and have them trained. BMDO has sent approximately 30 representatives through SCE training at SEI. Since most team members are unable to attend training at the same time, it is important to have a team perform a practice evaluation prior to the official evaluation. The practice allows them to become acquainted with each other and to determine the best approach for establishing a cohesive evaluation approach; e.g., allocate responsibilities, establish an interview style, ensure a common understanding of the evaluation criteria.

C.1.2 Review program results

A contractor answered questionnaire and multiple project profiles are used to select the most appropriate projects for the SCE team to evaluate at the contractor's site.

Review contractor questionnaires. The SEI questionnaire is usually submitted with the contractor's proposal. It is used by the SCE team to select projects with low process maturity and to identify Key Process Areas (KPAs) that may be weak. The BP SCE team however determined that the questionnaires did not typically reflect a contractor's process. It was very common for all of the answers to the questions to depict a level 5 process maturity as opposed to a more realistic level (i.e., level 2 or 3).

Select projects to be evaluated. The contractors provided profiles of numerous projects from which the team selected 2-3 to review in detail. Project profiles help the team identify the similarities and differences between the projects and to select the most appropriate ones to evaluate. The BP SCE team however only requested profiles on 3-5 projects. This was insufficient since many projects that were submitted were not similar to the BP program. The BE team however requested information on 7-9 projects which was found to be sufficient. It is important to ensure that the statements in the RFP provide adequate guidance to the contractors as to the types of projects they are to submit.

C.2 COLLECTING RISK DATA

C.2.1 Identify candidate risks via checklist

The SEI methodology requires the team to select a process maturity goal against which the competing contractors will be measured. Level 3 KPAs were used to evaluate all the competing contractors processes for the BP, BE, and NTF programs; i.e., project management, project planning, configuration management, quality assurance, standards and procedures, training, peer reviews, and software engineering process group. It is also practice to score the SEI questionnaire that is submitted with the proposals. The combination of the questionnaire responses and the project profile descriptions are used to help identify candidate risks to explore during the site visit.

C.2.2 Identify further data needs.

After selecting 2-3 projects to review in detail, the SCE team identifies additional data that is used in the evaluation. At least 1 week before the site visit, the team submits a request to the contractor containing an interview schedule with names of individuals from each project and a detailed documentation list for each project under review. The interview list contains about 30 names with time frames designated for the length of each interview. The documentation list contains project level documents (e.g., SDP, CM plan, QA plan) as well as organization level documents (e.g., standards, procedures, policies). These documents are to be made available upon the teams arrival at the contractor's facility.

In addition, a request is made for the contractor to give a presentation to the SCE team on their overall software development process as well as a description of each organization's responsibilities relative to the process. It is very important to provide the contractor the topics to be covered during this presentation to ensure the SCE team is not subjected to a sales pitch. It is also important to limit the presentation to two hours, otherwise the contractors may spend a day on their presentation and not allow the SCE team sufficient time to perform interviews and review documents.

C.2.3 Plan and conduct site visits

There are many activities associated with planning and conducting the site visit.

Delegate key process areas. Ideally it is beneficial if all the team members were experts in all of the KPAs and had sufficient time to evaluate each of them during the course of the 3-day SCE. But in practice, there is not enough time or expertise on the team. Thus, it has proven beneficial to assign each team member 1-2 KPAs. This helps to ensure all the areas are covered.

Develop evaluation checklist. The SCEs are based on the SEI Capability Maturity Model (CMM) which is about 400 pages long and infeasible to reference during an interview. Thus, the BP SCE teams developed interview questionnaires based on the CMM. The questionnaires however contained long cumbersome questions and the interviewers would progress through the questions without following-up on the responses. The best technique for conducting the interviews is to use a checklist that lists in an abbreviated form the software development practices the team is reviewing. The elimination of canned questions encouraged subsequent BMD SCE team members to interact with the interviewer and to understand the responses.

Establish interview technique. SEI teaches SCE teams to conduct one on many interviews. In other words, one contractor representative is interviewed by the 5-6 member SCE team. This approach is intimidating for the single contractor representative and frequently results in several more interviews than necessary because candidates are not necessarily familiar with the subject area. Rather than limit the contractor to one representative at a time, it may be more beneficial to allow the contractor to select individuals they feel are most qualified to explain the company's development process. But limit the contractor to about 4 people at a time in order to encourage questions and exchange of information.

Request detailed documentation. To ensure a process is actually in place and operating effectively, the SCE team requests detailed documentation during the course of the interviews. Examples of detailed documents that may be requested include minutes of the change control board meetings and checklists or errors reports from the peer review meetings. It would be beneficial to have a standard list of detailed documentation prepared for each KPA. This would help to ensure each team evaluates each KPA consistently and thoroughly.

Coordinate documents. Since numerous documents are requested during the course of the interviews it is important to establish a method to record what documents were requested and received and to ensure that the primary person on the SCE team reviews the document before it is returned to the contractor.

C.3 ASSESSING RISKS

C.3.1 Rank risks by probable impact

The SCE method can be used to evaluate and report risks to the source selection board or program management several different ways. The SCE results can be used to evaluate the accuracy of the contractor's questionnaire which was submitted with the proposal. If a contractor's questionnaire characterizes a level 5 organization and the SCE results indicate that the contractor is a level 1, this would be designated as high risk. Another approach is to use the SCE results to assess whether the contractor's process improvement plan was realistic. And a third method is to contrast the SCE results to the proposed software development process contained in the project SDP.

C.4 REPORTING FINDINGS

C.4.1 Integrate conclusions

Before conclusions can be generated, the SCE team must reach consensus and generate the findings for each KPA.

Reach consensus. The SCE team requires sufficient time during the course of the 3-day SCE to reach consensus on their findings. As the result of in-depth team discussions, the team may determine that additional interviews must be conducted or additional documents requested. It is best to assess each KPA during the visit to allow for additional interviews and document reviews as needs arise. But no conclusions are reported unless there is team consensus.

Evaluate each KPA. The SCE team evaluates each KPA as being acceptable or unacceptable and lists the findings associated with each KPA in terms of strengths, weaknesses, and planned process improvements. In practice it is unclear whether the final evaluation of each KPA should be done by the SCE team or the SSEB. The SCE team has a better understanding of the detailed findings where as the SSEB is responsible for scoring each contractor's results.

C.4.2 Brief findings

Findings are presented to both the government organization and the contractor.

Present findings to the SSEB or program manager. After each SCE, a member of the SCE team presents the findings to the SSEB or the program manager. The results are in the format of a presentation that is accompanied by a report. The presentation reviews the findings of each KPA and the report contains the details of the site visit. It is important to record the results of the interviews, document reviews, and the rational behind each of the findings. The report and the SCE team member's notebooks must be submitted to the SSEB since they are considered source selection sensitive during an acquisition process.

Present findings to the contractors. During the source selection process, contractors are not presented the results of the SCE until the losers conference which may be several months after an award is granted. It may be better practice to present the SCE findings to the contractor at the end of each site visit and to allow them an opportunity to comment on the findings and present proof that corrects any misinterpretations by the SCE team.

When SCEs are performed as a contract monitoring tool, the contractor is typically briefed at the end of the site visit.

C.5 PLANNING RISK REDUCTION

The SCE process does not include activities associated with reducing risks. After the SCE is complete, the results are provided to the SSEB or the program manager. It is not the role of the SCE team to mitigate the risks that were identified during the course of the SCE.

LIST OF REFERENCES

[AFSC 1988] Air Force Systems Command. 1988. *Software Risk Abatement.* Pamphlet AFSC/AFLCP 800-45. Washington, DC: Air Force Systems Command.

[Boehm 1989] Boehm, Barry. 1989. *Software Risk Management.* IEEE Computer Society Press. New York, NY: IEEE

[Carr 1993] Carr, Marvin J., et al. June 1993. *Taxonomy-Based Risk Identification.* CMU/SEI-93-TR-6. Pittsburgh, PA: Software Engineering Institute.

[DoD 1985] Department of Defense. September 1985. *Transition from Development to Production: Solving the Risk Equation.* DoD Manual 4245.7-M. Washington, DC.

[DoD 1988] Department of Defense. 29 February 1988. Military Standard DoD-STD-2167A, *Defense System Software Development.* Washington, DC.

[DoD 1992] Department of Defense. 22 December 1992. *Draft Military Standard Software Development and Documentation.* Washington, DC.

[Donis 1990] Donis, J. N. et al. January 1990. *Assessment of the Development Program for the AN/BSY-2 Submarine Combat System.* IDA Paper P-2355. Alexandria, VA: Institute for Defense Analyses.

[Donis 1993] Donis, J. N. et al. June 1993. *Technical Risk Indicators for Embedded Software Development.* IDA Paper P-2807. Alexandria, VA: Institute for Defense Analyses.

[EIA 1989] Electronic Industries Association. 1989. *DOD Computing Activities and Programs Ten-Year Market Forecast Issues, 1985-1995.* New York, NY.

[Humphrey 1990] Humphrey, Watts S. August 1990. *Managing the Software Process.* Reading MA: Addison-Wesley Publishing Co.

[IEEE 1993] Institute of Electrical and Electronic Engineers. February 1993. *Standard for Software Safety Plans.* Project P-1228 Draft J. New York, NY.

[JPL 1987] Jet Propulsion Laboratory. 1987. *Software Development: Formal Inspections*. Training Course Notes, Revision G. Pasadena, CA: California Institute of Technology.

[Jordano 1991] Jordano, Anthony, J. April 1991. *Software Development*. Briefing to IDA. Alexandria, VA: Institute for Defense Analyses.

[Kimmel 1993] Kimmel, H. Steven. August 1993. *The Plight of DoD Software Acquisition*. Washington, DC: Office of the Under Secretary of Defense (Acquisition and Technology).

[Paulk 1993] Paulk, Mark C. et al. February 1993. *Capability Maturity Model for Software, Version 1.1*. CMU/SEI-93-TR-24. Pittsburgh, PA:Software Engineering Institute.

[SEI 1993] Software Engineering Institute. March 1993. *Proceedings of the Second SEI Conference on Software Risk*. Pittsburgh, PA.

[Springsteen 1991] Springsteen, Beth. September 1991. *Software Maturity Model Applied to SDI*. IDA Document D-1042. Alexandria, VA: Institute for Defense Analyses.

[Springsteen 1992] Springsteen, Beth. September 1992. *Conducting Software Capability Evaluations*. IDA Paper P-2771. Alexandria, VA: Institute for Defense Analyses.

[Watson 1992] Watson, John, and David Harris. April 1, 1993. Revised Software Trust Principles. Program Information Report 93059. Falcon AFB, CO: Martin Marietta Corporation.

LIST OF ACRONYMS

AFSC	Air Force Systems Command
BE	Brilliant Eyes
BMDO	Ballistic Missile Defense Organization
BP	Brilliant Pebbles
CDRL	Contract Data Requirements List
CMM	Capability Maturity Model
COTS	Commercial off-the-shelf
CSCI	Computer Software Configuration Item
DEM/VAL	Demonstration and Validation phase of acquisition
DoD	Department of Defense
EMD	Engineering and Manufacturing Development phase
FFRDC	Federally funded research and development center
KPA	Key Process Area
OSD	Office of the Secretary of Defense
PM	Program Manager (of an acquisition program)
PO	Program Office (for an acquisition)
RFP	Request for Proposals
SCE	Software Capability Evaluation
SDD	Software Design Document
SDP	Software Development Plan
SEI	Software Engineering Institute (of Carnegie Mellon Univ.)
SOW	Statement of Work
SRS	Software Requirements Specification

SSS	System/Subsystem Specification
TEMP	Test and Evaluation Master Plan